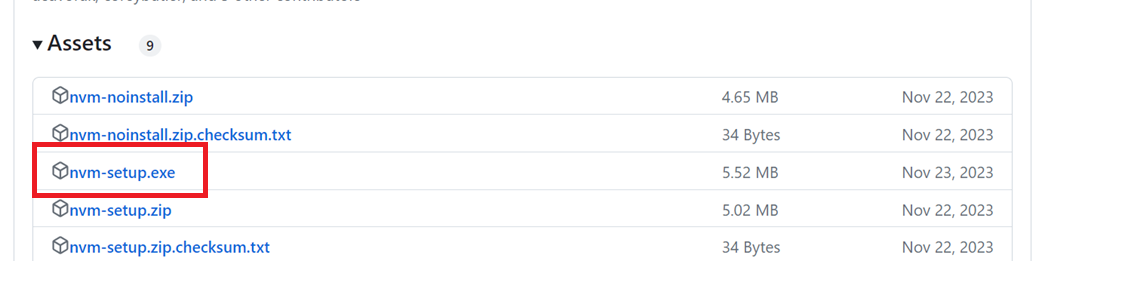
**NODEJS**

install nvm to manage nodejs version so that you can download older versions

[**https://github.com/coreybutler/nvm-windows/releases**](https://github.com/coreybutler/nvm-windows/releases)

****

**nvm ls-remote**

**nvm install 18.10.0**

**nvm use 18.10.0**

**node –version**

**if you want to download latest version of nodejs the LTS:** ((לא עובד גרסה אחרונה

[**https://nodejs.org/en/download/**](https://nodejs.org/en/download/)

backend nodejs api  with sql server

<https://www.youtube.com/watch?v=ErK3Qt52a1M&ab_channel=EducationwithAnkur>

A screenshot of a computer

Description automatically generated

1. Npm init –y
2. Npm install express mssql cors body parser
3. Npm install fs-extra
4. npm install nodemon –save-dev
5. npm install
6. add in package.json after line 8:

"start":"nodemon index.js"

1. npm install dotenv
2. create 3 folders: controllers,routes,data
3. create .env file
4. create file config.js
5. Write in .env file:

*NODE\_ENV=development*

*# express server config*

*PORT=8081*

*HOST=localhost*

*HOST\_URL=http://localhost:8081*

*# sql server config*

*SQL\_USER=sa*

*SQL\_PASSWORD=sql@123*

*SQL\_DATABASE=myDB*

*SQL\_SERVER=127.0.0.1 #You can check in computer management*

*SQL\_ENCRYPT=false*

1. Write in config.js file:

*'use strict';*

*const dotenv = require('dotenv');*

*const assert = require('assert');*

*dotenv.config();*

*const { PORT, HOST, HOST\_URL, SQL\_USER, SQL\_PASSWORD, SQL\_DATABASE, SQL\_SERVER } = process.env;*

*const sqlEncrypt = process.env.SQL\_ENCRYPT === "true";*

*assert(PORT, 'PORT is required');*

*assert(HOST, 'HOST is required');*

*module.exports = {*

*port:PORT,*

*host:HOST,*

*url:HOST\_URL,*

*sql: {*

*server:SQL\_SERVER,*

*database:SQL\_DATABASE,*

*user:SQL\_USER,*

*password:SQL\_PASSWORD,*

*options:{*

*encrypt:sqlEncrypt,*

*enableArithAbort:true*

*},*

*},*

*};*

1. Create a file myRoutes.js in routes folder and write into the file:

*'use strict';*

*const express = require('express');*

*const userControll = require('../controllers/userController');*

*const router = express.Router();*

*const {getAllUser,getUserById,createUser,deleteUserById,updateUser}=userControll*

*router.get('/users/getAllUser', getAllUser);//Send the all user details from the table*

*router.get('/users/getUserById/:id', getUserById);*

*router.post('/users/createUser',createUser)*

*router.delete('/users/deleteUserById/:id', deleteUserById);*

*router.put('/users/updateUser/:id',updateUser)*

*module.exports = {*

*routes: router*

*}*

1. Write in index.js file:

'use strict';

const express = require('express');

const config = require('./config');

const cors = require('cors');

const bodyParser = require('body-parser');

const userRoutes = require('./routes/myRoutes');

const app = express();

app.use(express.json());

app.use(cors());

app.use(bodyParser.json());

app.use('/api', userRoutes.routes);

app.listen(config.port, () => {

console.log('your app listening on url http://localhost:' + config.port )

});

1. Create new file utils.js in data folder and write into:

'use strict';

const fs = require('fs-extra');

const {join} = require('path');

const loadSqlQueries = async (folderName) => {

const filePath = join(process.cwd(), 'data', folderName);

const files = await fs.readdir(filePath);

const sqlFiles = files.filter(f => f.endsWith('.sql'));

const queries = {};

for (const sqlfile of sqlFiles) {

const query = fs.readFileSync(join(filePath, sqlfile), {encoding: "UTF-8"});

queries[sqlfile.replace(".sql", "")] = query;

}

return queries;

}

module.exports = {

loadSqlQueries

}

1. Create a file GetAllUser.sql into new folder users:

17.Create a file index.js and write:

'use strict';

const utils = require('../utils');

const config = require('../../config');

const sql = require('mssql');

const getUser = async () => {

try {

let pool = await sql.connect(config.sql);

const sqlQueries = await utils.loadSqlQueries('Users');

const userList = await pool.request()

.query(sqlQueries.GetAllUser);//This GetAllUser is Sql file name inside the data/users folder

return userList.recordset;

} catch (error) {

console.log(error.message);

}

}

module.exports = {

getUser

}

1. Create a file userController.js in controllers and write:

'use strict';

const userData = require('../data/users');

const getAllUser = async (req, res, next) => {

try {

const userlist = await userData.getUser();

res.send(userlist);

} catch (error) {

res.status(400).send(error.message);

}

}

const getUserById = async (req, res, next) => {

try {

const id = req.params.id;

const \_user = await userData.getUserById(id);

res.send(\_user);

} catch (error) {

res.status(400).send(error.message);

}

}

const createUser = async (req, res, next) => {

try {

const data = req.body;

const insert = await userData.createUser(data);

res.send(insert);

} catch (error) {

res.status(400).send(error.message);

}

}

const deleteUserById = async (req, res, next) => {

try {

const id = req.params.id;

const \_user = await userData.deleteUserById(id);

res.send(\_user);

} catch (error) {

res.status(400).send(error.message)

}

}

const updateUser = async (req, res, next) => {

try {

const id = req.params.id;

const data = req.body;

const updated = await userData.updateUser(id, data);

res.send(updated);

} catch (error) {

res.status(400).send(error.message);

}

}

module.exports = {

getAllUser,

getUserById,

createUser,

deleteUserById,

updateUser

}

19. npm install

20. npm start

21. http://localhost:8081/api/users/getAllUser